Capturing Crosslinguistic Generalizations: Multilingual Metagrammars

Tatjana Scheffler

Department of Linguistics, University of Pennsylvania

Swarthmore, March 6, 2007
Goals of This Talk

1. Give a brief overview of some aspects of computational linguistics
2. Discuss some recurring properties of languages
3. Present an approach that captures cross-linguistic generalizations
Outline

Linguistic Resources in Computational Linguistics
   What is Computational Linguistics?
   An Example Application of CL
   Multilingual Metagrammars

Two Cross-Linguistic Word Order Puzzles
   Scrambling
   The Verb-Second Constraint

A Multilingual Metagrammar
   Implementing Scrambling
   Implementing Verb-Second
   Sample Derivations

Conclusion
Outline

Linguistic Resources in Computational Linguistics

What is Computational Linguistics?
An Example Application of CL
Multilingual Metagrammars

Two Cross-Linguistic Word Order Puzzles
Scrambling
The Verb-Second Constraint

A Multilingual Metagrammar
Implementing Scrambling
Implementing Verb-Second
Sample Derivations

Conclusion
What is Computational Linguistics?

Theoretical Computational Linguistics

- formal theories of linguistic knowledge
- computational models of human cognition
- computational psycholinguistics

Applied Computational Linguistics

- human language technology / natural language processing
- human-machine interaction
- dealing with large corpora (internet)
- machine translation
Machine Translation (MT)

- A real-world example (German Historical Museum):

  (1) Königin Victoria aß gerne und viel.
      Queen Victoria ate with-pleasure and lots
  (2) Queen Victoria liked to eat and she ate a lot.

- A simpler example:

  (3) She likes to eat.  
  (4) Gerne isst sie.
      with-pleasure eats she

- What steps are needed to get from (3) to (4)?
  - identifying words, translating them
  - But looking up words is not enough!
MT – Different Methods of Transfer

- Analysis
- Direct Translation
- Generation

interlingua

source text → transfer → target text
MT – The Need for Grammars

- Independently of the translation strategy, idiosyncrasies of the source and target language have to be respected.

![Diagram showing tree structures for English and German sentences. The English sentence is: "She likes to eat." The German sentence is: "Sie isst gerne." The trees illustrate the constituent parts of the sentences.]
Grammars in Computational Linguistics

- Grammars describe the linguistic properties of a language in a concise way.
- In most CL applications, grammars are needed
  - hand-crafted grammars
  - grammars that have been extracted from (hand-crafted) corpora
- Developing such grammars is costly and slow.
Metagrammars

- *Metagrammars describe grammars*
- They contain partial descriptions of syntactic structure, which are compiled into actual grammars
- Elements of the syntactic descriptions can be explicitly reused:
  - within a grammar (e.g., properties of noun phrases, argument structures)
  - across grammars (this talk)
Motivation for Multilingual Metagrammars

Traditional focus: Grammar development
  ▶ guarantee consistency and coverage

Our focus: Linguistic generalizations
  ▶ develop new grammars for new languages quickly

Our approach: Find cross-linguistic and framework-neutral syntactic invariants
Cross-linguistic and cross-framework syntactic invariants

- Finite number of syntactic categories (NP, PP, etc.)
- Notion of subcategorization (intransitive, transitive, etc.)
- Finite number of syntactic functions (subject, object etc.)
- Existence of valency alternations (passive, causative, etc.)
- Argument realization, word order effects (such as V2 or wh-movement)
Outline

Linguistic Resources in Computational Linguistics
  What is Computational Linguistics?
  An Example Application of CL
  Multilingual Metagrammars

Two Cross-Linguistic Word Order Puzzles
  Scrambling
  The Verb-Second Constraint

A Multilingual Metagrammar
  Implementing Scrambling
  Implementing Verb-Second
  Sample Derivations

Conclusion
Scrambling in Korean

- Korean is a verb-final language with relatively free word order.
- Noun Phrases exhibit *scrambling*.
- *Scrambling* is the permutation of constituents (arguments, adjuncts).

(5) \[[\text{hyeongi}_\text{gongjangi}]_1 [\text{samchonege}]_2 [\text{gagureul}]_3 \]

\[\text{a}_{\text{local\_company}} \text{nom \ the\_uncle}_{\text{dat}} \text{\ furniture}_{\text{acc}} \]

\[\text{[samiljeone]}_4 \text{baedakhaessda.} \]

three\_days\_ago\ delivered\_has.

‘A local company has delivered the furniture to the uncle three days ago’

- $4! = 24$ word orders are acceptable for this sentence in Korean.
Scrambling in German

- German is another SOV language with scrambling.

(6) ... (dass) [eine hiesige Firma]₁ [dem Onkel]₂ [die Möbel]₃ [vor drei Tagen]₄ zugestellt hat.

... (dass) [vor drei Tagen]₄ [dem Onkel]₂ [eine hiesige Firma]₁ [die Möbel]₃ zugestellt hat.

... (dass) [die Möbel]₃ [dem Onkel]₂ [vor drei Tagen]₄ [eine hiesige Firma]₁ zugestellt hat.

... (dass) [dem Onkel]₂ [vor drei Tagen]₄ [eine hiesige Firma]₁ [die Möbel]₃ zugestellt hat.

... (dass) [dem Onkel]₂ [vor drei Tagen]₄ [eine hiesige Firma]₁ [die Möbel]₃ zugestellt hat.

... that a local company₁ has delivered the furniture₃ to the uncle₂ three days ago₄.
The Verb-Second Phenomenon (V2)

(7) a. [Auf dem Weg] **sieht** [der Junge] [eine Ente].
    on the path sees the boy a duck
    ‘On the path, the boy sees a duck.’

b. * [Auf dem Weg] [der Junge] **sieht** [eine Ente].
    on the path the boy sees a duck
    Int.: ‘On the path, the boy sees a duck.’

- Finite verb is required to be located in “second position”
- V2 languages include German, Dutch, Yiddish, Frisian, Icelandic, Mainland Scandinavian, and Kashmiri
- Small-scale linguistic variation: Behavior in embedded clauses differs
V2 in German

(8) a. Der Junge **sieht** eine Ente auf dem Weg.
the boy **sees** a duck on the path
‘On the path, the boy sees a duck.’

b. . . ., dass der Junge auf dem Weg eine Ente **sieht**.
. . ., that the boy on the path a duck sees
‘. . ., that the boy sees a duck on the path.’

- Main clauses exhibit V2 in German
- Embedded clauses with complementizers are verb-final

<table>
<thead>
<tr>
<th>German</th>
<th>Main Clauses</th>
<th>Embedded Clauses</th>
</tr>
</thead>
<tbody>
<tr>
<td>V2</td>
<td>V-Final</td>
<td></td>
</tr>
</tbody>
</table>
A First Explanation of German Word Order

- German is a verb-final language.
- In main clauses, the verb moves to the complementizer position, and some constituent *topicalizes* (moves) to its specifier.

```
CP
  PP   C’
   on the path
  C
    V   NP
      sees the boy NP
       Obj a duck V
         t
```
In embedded clauses, the overt complementizer blocks this.

\[
\begin{align*}
CP & \quad C' \\
C & \quad \text{that} \\
\text{NP}_{\text{Subj}} & \quad \text{the boy} \\
\text{NP}_{\text{Obj}} & \quad \text{a duck} \\
V' & \quad \text{sees}
\end{align*}
\]
V2 in Yiddish

(9) a. Oyfn veg zet dos yingl a katshke. on-the path sees the boy a duck.‘On the path, the boy sees a duck.’
b. . . . , az dos yingl zet a katshke oyfn veg . . . , that the boy sees a duck on-the path ‘. . . , that the boy sees a duck on the path.’

▶ As a verb-second language, Yiddish main clauses exhibit V2
▶ Yiddish embedded clauses must also be V2

<table>
<thead>
<tr>
<th></th>
<th>Main Clauses</th>
<th>Embedded Clauses</th>
</tr>
</thead>
<tbody>
<tr>
<td>German</td>
<td>V2</td>
<td>V-Final</td>
</tr>
<tr>
<td>Yiddish</td>
<td>V2</td>
<td>V2</td>
</tr>
</tbody>
</table>
Summary: Two Puzzles

1. Scrambling
   - free reordering of constituents in Korean, German, . . .

2. Verb-Second Constraint
   - finite verb in second position in main clauses
   - but in embedded clauses, the behavior differs
   - What is the cross-linguistic core of this phenomenon?
Outline

Linguistic Resources in Computational Linguistics
  What is Computational Linguistics?
  An Example Application of CL
  Multilingual Metagrammars

Two Cross-Linguistic Word Order Puzzles
  Scrambling
  The Verb-Second Constraint

A Multilingual Metagrammar
  Implementing Scrambling
  Implementing Verb-Second
  Sample Derivations

Conclusion
Some Assumptions

- We are working with Tree-Adjoining Grammar (not introduced here).
- All verbal phrasal nodes are called VP, they will be distinguished by certain features.
  - This is necessary to capture freer word order.
  - Continuation of the distinction between V’, VP, I’, IP, C’, CP, etc.
- Modifiers are not currently part of the (meta)grammar.
Scrambling in the Metagrammar

- Free order through underspecification

Metagrammar:

(Compiled) Grammar:
Idea  Basic V2 phenomenon is the same in all V2 languages: Topicalization

Our Approach  Crosslinguistic generalizations are captured in one Metagrammar using different heads (verbs) (see Rambow and Santorini, 1995)
Major New Issue When Going Multilingual: Heads

- One language: relative position of verb and arguments determine word order
- Two languages: want language-independent generalizations about syntax; prototypical example: adverbs in English and French (Pollock):
  E: Charles (often) eats (*often) beans
  F: Charles (souvent) mange (souvent) des haricots
- Solution: verbal heads are in different positions on the projection in E and F, but adverb is always adjoined to the left of VP
- In some languages (like German and Yiddish), it is clear that verbs can be in different positions on the projection, anyway
- For some languages (Korean), there is very little evidence for this notion
Dealing With Word Order Variation in a Metagrammar

Verbal trees are determined by:

1. A subcategorization frame (e.g., intransitive/transitive)
2. Valency alternations (e.g., active/passive)
3. Argument realizations (e.g., wh-movement)
4. A topology, which encodes the position and characteristics of the verbal head

```
that

the boy

a duck

sees
```
Topology

A topology is a combination of the projection and any compatible head(s).

projection

- Empty verbal head plus its maximal projection
- Different types of clauses defined by features:
  - non-finite clauses: [I:-]
  - root V2 clauses: [Top:+]
  - finite clauses [M:+, I:+]

heads

- Introduce categorial features
- The list of possible heads differs from language to language
A Finite Projection

\[
\begin{array}{c}
\text{VP} \\
\text{M} \\
\text{I} \\
\text{VP} \\
\text{C} \\
\text{M} \\
\text{TOP} \\
\text{I} \\
\varepsilon
\end{array}
\]
The Heads Define the Topology of Clauses

Properties of the verbal heads (feature inventory) determine the positions of arguments and adjuncts:

- **I** (finite tense and subject-verb agreement): creates a specifier position for agreement, but allows recursion (i.e., adjunction at IP)
- **Top** (topic): a feature which creates a specifier position for the topic and which does not allow recursion (used for V2)
- **M** (mood): a feature with semantic content (to be defined), but no specifier
- **C** (complementizer): a lexical feature introduced only by complementizers
Some Simplified German Heads

1:  \[ \text{VP} \left[ \begin{array}{c} \text{C} \\ \text{M} \\ \text{TOP} \\ \text{I} \end{array} \right] \text{ - } \]  

2:  \[ \text{VP} \left[ \begin{array}{c} \text{C} \\ \text{M} \\ \text{TOP} \\ \text{I} \end{array} \right] \text{ - } \]  

4:  \[ \text{VP} \left[ \begin{array}{c} \text{C} \\ \text{M} \\ \text{TOP} \\ \text{I} \end{array} \right] \text{ - } \]  

finite V-final  \quad V2-Subject  \quad \text{Complementizer}
## German vs. Yiddish Heads

### German:

<table>
<thead>
<tr>
<th>What</th>
<th>Features Introduced</th>
<th>Directionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Verb (clause-final)</td>
<td>+I</td>
<td>head-final</td>
</tr>
<tr>
<td>2 Verb (V2, subject-initial)</td>
<td>+M, +Top, +I</td>
<td>head-initial</td>
</tr>
<tr>
<td>3 Verb (V2, non-subject-initial)</td>
<td>+M, +Top</td>
<td>head-initial</td>
</tr>
<tr>
<td>4 Complementizer</td>
<td>+C, +M</td>
<td>head-initial</td>
</tr>
</tbody>
</table>

### Yiddish:

<table>
<thead>
<tr>
<th>What</th>
<th>Features Introduced</th>
<th>Directionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Verb</td>
<td>+I</td>
<td>head-initial</td>
</tr>
<tr>
<td>2 Verb (V2, subject-initial)</td>
<td>+M, +Top, +I</td>
<td>head-initial</td>
</tr>
<tr>
<td>3 Verb (V2, non-subject-initial)</td>
<td>+M, +Top</td>
<td>head-initial</td>
</tr>
<tr>
<td>4 Complementizer</td>
<td>+C</td>
<td>head-initial</td>
</tr>
</tbody>
</table>
Derivation of a German sentence

```
Derivation of a German sentence

VP [C + ]
  [M + ]
  [I + ]

comp

VP [C - ]
  [M - ]
  [I + ]

NP_{Obj}   VP

NP_{Subj}  VP [I + ]

VP

VP [M + ]
  [I + ]

VP

Head 4 (Comp) + Object-Non-Topicalized + Subject-Non-Topicalized + Projection + Head 1 (V-final)

that

a duck

a boy

saw

Tatjana Scheffler (UPenn)  Multilingual Metagrammars  Swarthmore, March 6, 2007
Derived German Tree

```
VP [ C:+, M:+, Top:−, I:+ ]
  /    
 comp   VP [ C:−, M:−, Top:−, I:+ ]
     /    
 NP Obj  VP [ C:−, M:−, Top:−, I:+ ]
   /    
 NP Subj VP [ C:−, M:−, Top:−, I:+ ]
     /    
   VP [ C:−, M:−, Top:−, I:+ ]
     /    
    ε
```
Derivation of a Yiddish Sentence
Derived Yiddish Tree

```
VP [ C:+, M:+, l:+ ]
  comp
  VP [ C:−, M:+, l:+ ]
    NP_{Subj}
    VP [ C:−, M:+, Top:+, l:+ ]
      v
    VP [ C:−, M:−, Top:+, l:+ ]
    VP [ C:−, M:−, Top:+, l:+ ]
    NP_{Obj}
    ε
```
Outline

Linguistic Resources in Computational Linguistics
  What is Computational Linguistics?
  An Example Application of CL
  Multilingual Metagrammars

Two Cross-Linguistic Word Order Puzzles
  Scrambling
  The Verb-Second Constraint

A Multilingual Metagrammar
  Implementing Scrambling
  Implementing Verb-Second
  Sample Derivations

Conclusion
Conclusion

- Grammars are needed in virtually all CL applications
- Metagrammar captures common elements in and among grammars
- Ideal for representing cross-linguistic generalizations
- Korean, German and Yiddish look a lot alike in a metagrammar
- Very fast development of grammars for new languages is possible
Thank You!
Universal Grammar components

- A clausal tree is defined by a projection, a subcategorization frame, and a set of heads
- Category of the arguments, for example, is underspecified in the UG
- Head and its sister are not ordered in UG (double arrow)

Generic elements of Universal Grammar: projection, head, argument (from left to right)
UG components (2)

- Spec heads, non-spec heads
- specifier arguments, non-specifier arguments
- universal diathesis alternations: passive, causative